**ls, revisited**

So far we’ve used the command line to navigate the filesystem.

We can do more with the command line to view directories and files. We can also use the command line to copy, move, and remove files and directories. Let’s see how to do this.

You can reference the filesystem for this lesson [here](https://s3.amazonaws.com/codecademy-content/courses/learn-the-command-line/img/LCL-fileTrees-02.png). It is highly recommended for this lesson.

**Instructions**

**1.**

In the terminal after the shell prompt, type

ls

**2.**

Then type

ls -a

Do you see the differences between the outputs of both commands? Click Next to learn how this works.

# ls -a

$ ls -a

. .. .preferences action drama comedy genres.txt

1. The ls command lists all files and directories in the working directory.
2. The -a modifies the behavior of the ls command to also list the files and directories starting with a dot (.). Files started with a dot are hidden, and don’t appear when using ls alone.

The -a is called an option. Options modify the behavior of commands. Here we used ls -a to display the contents of the working directory in more detail.

In addition to -a, the ls command has several more options. Here are three common options:

* -a - lists all contents, including hidden files and directories
* -l - lists all contents of a directory in long format
* -t - order files and directories by the time they were last modified.

Let’s practice using these options below.

**Instructions**

**1.**

In the terminal, type

ls -l

Click Next to find out what these columns mean.

**ls -l**

$ ls -l

drwxr-xr-x 5 cc eng 4096 Jun 24 16:51 action

drwxr-xr-x 4 cc eng 4096 Jun 24 16:51 comedy

drwxr-xr-x 6 cc eng 4096 Jun 24 16:51 drama

-rw-r--r-- 1 cc eng 0 Jun 24 16:51 genres.txt

The -l option lists files and directories as a table. Here there are four rows, with seven columns separated by spaces. Here’s what each column means:

1. Access rights. These are actions that are permitted on a file or directory.
2. Number of hard links. This number counts the number of child directories and files. This number includes the parent directory link (..) and current directory link (.).
3. The username of the file’s owner. Here the username is cc.
4. The name of the group that owns the file. Here the group name is eng.
5. The size of the file in bytes.
6. The date & time that the file was last modified.
7. The name of the file or directory.

**Instructions**

**1.**

Let’s try out another option for the ls command.

Navigate to the **comedy/** directory.

**2.**

Then type

ls -alt

# ls -alt

$ ls -alt

drwxr-xr-x 4 cc eng 4096 Jun 29 12:22 .

-rw-r--r-- 1 cc eng 0 Jun 29 12:22 .gitignore

drwxr-xr-x 5 cc eng 4096 Jun 30 14:20 ..

drwxr-xr-x 2 cc eng 4096 Jun 29 12:22 satire

drwxr-xr-x 2 cc eng 4096 Jun 29 12:22 slapstick

-rw-r--r-- 1 cc eng 14 Jun 29 12:22 the-office.txt

The -t option orders files and directories by the time they were last modified.

In addition to using each option separately, like ls -a or ls -l, multiple options can be used together, like ls -alt.

Here, ls -alt lists all contents, including hidden files and directories, in long format, ordered by the date and time they were last modified.

**Instructions**

**1.**

Let’s move on to copying, moving, and removing files and directories from the command line.

Navigate to the **drama/biopic/** directory.

cd ../drama/biopic/

List all files and directories in the working directory.

**2.**

Then type

cp frida.txt lincoln.txt

Click Next to learn about this command.

# cp I

cp frida.txt lincoln.txt

The cp command copies files or directories. Here, we copy the contents of **frida.txt** into **lincoln.txt**.

**Instructions**

**1.**

Navigate to the **drama/** directory.

cd ..

List all files and directories in the working directory.

**2.**

Type

cp biopic/cleopatra.txt historical/

**3.**

Navigate to the **historical/** directory.

List all files and directories in the working directory. You should see a new copy of **cleopatra.txt** in this directory.

**4.**

Here’s one more way to use cp.

Navigate up one directory from **drama/historical/** to **drama/**.

Stuck? Get a hint

**5.**

Then type

cp biopic/ray.txt biopic/notorious.txt historical/

**6.**

Change directories into **historical/**.

List all files and directories in the working directory. You should see a new copy of **ray.txt** and **notorious.txt** in this directory.

# cp II

cp biopic/cleopatra.txt historical/

To copy a file into a directory, use cp with the source file as the first argument and the destination directory as the second argument. Here, we copy the file **biopic/cleopatra.txt** and place it in the **historical/** directory.

cp biopic/ray.txt biopic/notorious.txt historical/

To copy multiple files into a directory, use cp with a list of source files as the first arguments, and the destination directory as the last argument. Here, we copy the files **biopic/ray.txt** and **biopic/notorious.txt** into the **historical/** directory.

**Instructions**

**1.**

Let’s look at two more ways to use cp.

Navigate to the **comedy/** directory.

cd ../../comedy

**2.**

In this directory, create a new file named **shrek.txt**. ([Here’s a hint](https://www.codecademy.com/en/courses/learn-the-command-line/lessons/navigation/exercises/touch) on how to do this.)

**3.**

Then type

cp \* satire/

**4.**

Navigate to the **satire/** directory.

List all files and directories in the working directory.

You should see a copy of the files **the-office.txt** and **shrek.txt** in this directory. We’ll explain how this works in the next exercise.

**5.**

Here’s another way to use cp.

Navigate to the **action/** directory. Type

cd ../../action/

Here we navigate up two directories, and then into the **action/** directory.

**6.**

Type

cp m\*.txt scifi/

**7.**

Change directories into **scifi/**.

List all files and directories in the working directory.

You should see a copy of all text files starting with “m”: **matrix.txt**, **matrix-reloaded.txt**, and **matrix-revolutions.txt**.

Click Next to learn how this works.

$ cd ../../comedy

bash: cd: ../../comedy: No such file or directory

$ la

bash: la: command not found

$ pwd

/home/ccuser/workspace/movies

$ ls

action comedy drama genres.txt

$ cd comedy/

$ touch shrek.txt

$ cp \* satire/

cp: -r not specified; omitting directory 'satire'

cp: -r not specified; omitting directory 'slapstick'

$ cd satire/

$ cd ../../action/

$ cp m\*.txt scifi/

$ ls

batman.txt matrix-revolutions.txt scifi superman.txt

matrix-reloaded.txt matrix.txt superhero wonderwoman.txt

$ cd scifi/

# Wildcards

cp \* satire/

In addition to using filenames as arguments, we can use special characters like \* to select groups of files. These special characters are called wildcards. The \* selects all files in the working directory, so here we use cp to copy all files into the **satire/** directory.

cp m\*.txt scifi/

Here, m\*.txt selects all files in the working directory starting with “m” and ending with “.txt”, and copies them to **scifi/**.

**Instructions**

**1.**

In addition to copying files, we can move files from the command line.

Change directories into the **action/** directory.

cd ../

**2.**

Type

mv superman.txt superhero/

**3.**

Navigate to the **superhero/** directory.

List all files and directories in the working directory. You should see **superman.txt** in it.

**4.**

Here’s another way to use mv.

Navigate up one directory from **action/superhero/** to **action/**.

**5.**

Then type

mv wonderwoman.txt batman.txt superhero/

**6.**

Navigate to **superhero/** again.

List all files and directories in the working directory. You should see **wonderwoman.txt** and **batman.txt** in it.

**7.**

Here’s one more way to use mv.

Type

mv batman.txt spiderman.txt

**8.**

List all files and directories in the working directory. You should see the file **batman.txt** has been renamed as **spiderman.txt**.

Click Next to learn how these commands work.

# mv

The mv command moves files. It’s similar to cp in its usage.

mv superman.txt superhero/

To move a file into a directory, use mv with the source file as the first argument and the destination directory as the second argument. Here we move **superman.txt** into **superhero/**.

mv wonderwoman.txt batman.txt superhero/

To move multiple files into a directory, use mv with a list of source files as the first arguments, and the destination directory as the last argument. Here, we move **wonderwoman.txt** and **batman.txt** into **superhero/**.

mv batman.txt spiderman.txt

To rename a file, use mv with the old file as the first argument and the new file as the second argument. By moving **batman.txt** into **spiderman.txt**, we rename the file as **spiderman.txt**.

**Instructions**

**1.**

Change directory to **comedy/slapstick**.

List all files and directories in the working directory.

cd ../../comedy/slapstick/

**2.**

Type

rm waterboy.txt

**3.**

List all files and directories in the working directory. You should see that **waterboy.txt** has been removed.

**4.**

Navigate up one directory from **comedy/slapstick/** to **comedy/**.

**5.**

Type

rm -r slapstick

**6.**

List all files and directories in the working directory. You should see that the **slapstick/** directory has been removed.

Click Next to learn how the rm command works.

Congratulations! You learned how to use the command line to view and manipulate the filesystem. What can we generalize so far?

* Options modify the behavior of commands:
  + ls -a lists all contents of a directory, including hidden files and directories
  + ls -l lists all contents in long format
  + ls -t orders files and directories by the time they were last modified
  + Multiple options can be used together, like ls -alt
* From the command line, you can also copy, move, and remove files and directories:
  + cp copies files
  + mv moves and renames files
  + rm removes files
  + rm -r removes directories
* Wildcards are useful for selecting groups of files and directories